Attorney Docket No.: 21751-005600US

PATENT APPLICATION

SAWTOOTH SPLINE DISPLAY

Inventors: Robert G. Podesta, a citizen of The United States, residing at
70 Rio Vista Avenue
Oakland, CA 94611

Scott P. Clark, a citizen of The United States, residing at
283 Clinton Park
San Francisco, CA 94103

Christopher R. Schoeneman, a citizen of The United States, residing at
91 Eucalyptus Road
Berkeley, CA 94705

Timothy S Milliron, a citizen of The United States, residing at
1417 Kains Avenue
Berkeley, CA 94702

Assignee: Pixar
1200 Park Avenue
Emeryville, CA, 94608

Entity: Large

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 650-326-2400

SAWTOOTH SPLINE DISPLAY

CROSS-REFERENCES TO RELATED APPLICATIONS

[0001] NOT APPLICABLE

STATEMENT AS TO RIGHTS TO INVENTIONS MADE UNDER FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] NOT APPLICABLE

REFERENCE TO A "SEQUENCE LISTING," A TABLE, OR A COMPUTER PROGRAM LISTING APPENDIX SUBMITTED ON A COMPACT DISK.

[0003] NOT APPLICABLE

BACKGROUND OF THE INVENTION

[0004] This invention relates to computer animation and more particularly to the use and display of an animation spline, namely the mathematically defined curve in distance and time that is used to produce a smooth path between successive points or key frames in an animation sequence. This invention addresses the problem of intuitively assessing and controlling ease in and ease out for key frames in an animation scene (also known as "slow in and out"). This invention is particularly relevant for monotonically increasing curves, such as those used to control the timing of other animation splines.

[0005] With the wide-spread availability of computers, film animators increasingly rely upon computers to assist in the animation process. This process includes using computers to facilitate drawing-based animation, for example, by painting images, by generating in-between images ("tweening"), and the like. This process also includes using computers to augment physical animation techniques. For example, physical models could be represented by virtual models in computer memory, and then the virtual models could be manipulated.

[0006] As one of the pioneering companies in the computer aided animation (CAA), Pixar of Emeryville, California, has developed computing platforms especially designed for CAA, animation software, and rendering software now known as RenderMan®. While

RenderMan® was focused upon rendering, namely the creation of images from geometric models, the animation software developed for in-house use had focused upon allowing animators to specify ("animate") the geometric models. The geometric models typically represent objects in a scene, characters in a scene, positions of objects and characters, manipulation of objects and characters, lighting, textures, and the like.

[0007] In the process of preparing a fully animated scene, animation splines are provided in display form to guide the animator in making dynamic changes between key frames. Animation splines are typically drawn against a constant distance axis relative to a time axis, typically a vertical axis. As such, an ease in/out effect for a monotonically increasing section takes on an S-curve shape through the knot. This S-curve is non-symmetrical about the vertical axis, making it more difficult for the animator to observe differences between the in and out sections, and as such it is also not especially intuitive to an animator. A further disadvantage of this approach is that splines that represent monotonically increasing quantities over an extended time period require a large amount of vertical space to display fully.

[0008] What is needed is a mechanism for enhancing the usefulness of animation splines.

SUMMARY OF THE INVENTION

[0009] According to the invention, in a computer aided animation system where an animation spline is produced from a sequence of knots, the display of the animation spline is modified so that the vertical axis of the curve is flipped at selected 'pose' knots in order to allow ready evaluation, typically by visual inspection, of the perception effect of the spline.

[0010] In order to more fully understand the present invention, reference is made to the accompanying drawings. Understanding that these drawings are not to be considered limitations in the scope of the invention, the presently described embodiments and the presently understood best mode of the invention are described with additional detail through use of the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Fig. 1 is a block diagram of an animation system according to one embodiment of the present invention.

[0012] Fig. 2 is a representation of an a pose spline in the ascending or unflipped mode.

[0013] Fig. 3 is a representation of the pose spline of Fig. 2 in the sawtooth or flipped mode.

[0014] Fig. 4 is a representation of an animation sequence, such as a vehicle along a path, from which a spline is constructed, using pose knots.

DESCRIPTION OF SPECIFIC EMBODIMENTS OF THE INVENTION

[0015] Fig. 1 is a block diagram of typical computer rendering system 100 according to an embodiment of the present invention.

[0016] In the present embodiment, computer system 100 typically includes a monitor 110, computer 120, a keyboard 130, a user input device 140, a network interface 150, and the like.

[0017] In the present embodiment, user input device 140 is typically embodied as a computer mouse, a trackball, a track pad, wireless remote, and the like. User input device 140 typically allows a user to select objects, icons, text and the like that are to appear on the monitor 110. In the present invention , an active spline is displayed as a two dimensional graph on the monitor 110 as hereinafter explained.

[0018] Embodiments of network interface 150 typically include an Ethernet card, a modem (telephone, satellite, cable, ISDN), (asynchronous) digital subscriber line (DSL) unit, and the like. The network interface 150 is typically coupled to a computer network. In other embodiments, the network interface 150 may be physically integrated on the motherboard of computer 120, may be a software program, such as soft DSL, or the like.

[0019] The computer 120 typically includes components such as a processor 160, and memory storage devices, such as a random access memory (RAM) 170, disk drives 180, and a system bus 190 interconnecting the above components.

[0020] In one embodiment, the computer 120 is a PC compatible computer having one or more microprocessors such as Xeon™ microprocessor from Intel Corporation. Further, in the present embodiment, the computer 120 may include a UNIX-based operating system.

[0021] RAM 170 and disk drive 180 are examples of tangible media for storage of data, audio/video files, computer programs, applet interpreters or compilers, virtual machines, embodiments of the herein described invention including an animation engine, geometric

description of objects, characters, object data files, character rigging, shader descriptors, a rendering engine, output image files, texture maps, displacement maps, scattering lengths and absorption data of object materials, a spline description, and the like. Other types of tangible media include magnetic disks such as floppy disks and Zip® disks, removable hard disks, optical storage media such as CD-ROMS and bar codes, semiconductor memories such as flash memories, read-only-memories (ROM), battery-backed volatile memories, networked storage devices, and the like.

[0022] In the present embodiment, computer system 100 may also include software that enables communication via a network such as according to HTTP, TCP/IP, RTP/RTSP protocols, and the like in order to communicate information during the work process as well as communicate a final product for storage and production. In alternative embodiments of the present invention, other communication software and transfer protocols may also be used, for example IPX, UDP or the like.

[0023] Fig.1 is representative of computer animation systems capable of embodying the present invention. It will be readily apparent to one of ordinary skill in the art that many other hardware and software configurations are suitable for use with the present invention. For example, the use of other micro processors are contemplated, such as Pentium™ or Itanium™ microprocessors; Opteron™ or AthlonXP™ microprocessors from Advanced Micro Devices, Inc; PowerPC G3™, G4™ microprocessors from Motorola, Inc.; and the like. Further, other types of operating systems are contemplated, such as Windows® operating system such as WindowsXP®, WindowsNT®, or the like from Microsoft Corporation, Solaris from Sun Microsystems, LINUX, UNIX, MAC OS and other operating systems from Apple Computer Corporation, and the like.

[0024] Referring to Fig. 4, an animation sequence is illustrated. A vehicle at position or pose knot "e" has traversed a path from position "a" through positions "b," "c" and "d," called poses, the mathematically equivalent nodes of the poses being called pose knots, along a segmented path. The distance between the poses in split, namely the sequence illustrated, is reflected in the distance between the knots in split. The greater the distance between the poses along the path, the greater will be the vertical separation between the pose knots when viewed graphically as a spline.

[0025] In embodiments of the present invention, the sawtooth spline and modifications to objects are implemented by the computer-aided animation system by producing a spline

between points "a" through "e" and displaying the spline on the display monitor 110 to the animator for evaluation. There are two ways an animator can evaluate the spline: the conventional technique wherein an ascending view of the spline is produced and displayed such that every successive pose knot is further along the distance scale as time is advanced, and a sawtooth mode wherein alternating pose knots are displayed with the vertical (distance) axis flipped. The animator thus has access to both the display of Fig. 2 and of Fig. 3 for an animation of Fig. 4 to adjust the spline trajectory through knots b, c and d, so that distinctions can be noted, and a smoother spline curve can be constructed for the animation sequence. The animator has the latitude of moving the pose knots in time. The bidirectional arrows of Figure 4 correspond to bidirectional arrows at respective pose knot positions for Figures 2 and 3.

[0026] According to the invention, a flexible method for successively flipping the vertical (distance) axis of an animation spline is provided. Referring to Fig. 3, the flipped mode of the spline of Fig. 2 is illustrated wherein the vertical (distance) axis is locally flipped through alternating knots. The visual effect is an exaggeration of anomalies in the curve as segments successively alternate direction reminiscent of a sawtooth waveform. Hence it is called a sawtooth spline.

[0027] This technique is particularly useful for managing splines that control timing of other curves. For example, the animator can adjust selected splines in time (left and right in Figs. 2 and 3) without impacting distance. The actual values of the knots are not affected. It is recognized that it may not be desirous to flip every successive knot in a spline. To facilitate the description of the invention, therefore, two types of knots are defined. The term 'pose' knot is adopted to identify a (spline) knot where a flip or axis reversal occurs, and the term 'timing' knot is adopted to refer to a knot where no change or flipped state occurs. In the algorithm, it will be noted that, if a sequence of pose knots exist with the same value, it would be most intuitive to flip only the vertical axis on the last pose knot in the sequence.

[0028] According to the invention, a sawtooth display may be implemented as a mode in the spline evaluation display of a computer-assisted animation system that can be switched between sawtooth mode (on) or ascending mode (off), as respectively shown in Fig. 3 and Fig. 2. When turned on, the display is kept up to date by keeping the flipped state of each knot in the spline up to date using the following algorithm, expressed in pseudo code.

[0029] inputs: the spline, the 'flipBeforeFirstPoseKnot' setting

[0030] 1. initialize 'flipped' flag to 'flipBeforeFirstPoseKnot'

[0031] 2. initialize 'nextFlip' to 0

[0032] 3. for every knot in the spline do

[0033] a. if the knot is a 'pose' knot, then

[0034] i. set 'flipped' to 'nextFlip'

[0035] ii. toggle 'nextFlip'

[0036] b. set knot's 'knotFlipped' flag to 'flipped'

[0037] c. set knot's 'firstPose' to 0

[0038] 4. initialize 'zeroPoint' to 0.0

[0039] 5. for every knot in the spline do

[0040] a. if the knot is a 'pose' knot, then

[0041] i. set knot's 'firstPose' to 1

[0042] ii. if 'flipBeforeFirstPoseKnot' is on, then

[0043] 1. set 'zeroPoint' to (2 * knot value)

[0044] iii. break out of for loop

[0045] 6. for every knot in the spline do

[0046] a. if the knot is a 'pose' knot, then

[0047] i. if this is the first pose knot ('firstPose' is 1), then

[0048] 1. set 'zeroPoint' to 0

[0049] ii. else, if this knot is flipped ('knotFlipped' is 1), then

[0050] 1. add (2 * knot value) to 'zeroPoint'

[0051] iii. else

[0052] 1. subtract (2 * knot value) from 'zeroPoint'

[0053] b. set the knot's 'knotZeroPoint' to 'zeroPoint'

[0054] Then, when displaying a spline, the y-coordinate for each knot is adjusted using the following algorithm.

[0055] inputs: the knot, the y-coordinate 'y', 'leftOfKnot', 'flipBeforeFirstPoseKnot'

[0056] output: the adjusted y-coordinate

[0057] 1. if not in sawtooth mode, then

[0058] a. return 'y' unchanged

[0059] 2. else

[0060] a. if 'leftOfKnot' is false

[0061] i. set 'zeroPoint' to the knot's 'knotZeroPoint'

[0062] ii. set 'flip' to the knot's 'flipped'

[0063] b. else, if there is a previous knot in the spline

[0064] i. set 'zeroPoint' to the previous knot's 'knotZeroPoint'

[0065] ii. set 'flip' to the previous knot's 'flipped'

[0066] c. else, if the knot is a pose knot

[0067] i. if 'flipBeforeFirstPoseKnot' is on

[0068] 1. set 'zeroPoint' to (2 * knot value)

[0069] ii. else

[0070] 1. set 'zeroPoint' to 0

[0071] iii. set 'flip' to 'flipBeforeFirstPoseKnot'

[0072] d. else,

[0073] i. set 'zeroPoint' to the knot's 'knotZeroPoint'

[0074] ii. set 'flip' to 'flipBeforeFirstPoseKnot'

[0075] e. if 'flip' is set to 1

[0076] i. return ('zeroPoint' - 'y')

[0077] f. else

[0078] i. return ('zeroPoint' + 'y')

[0079] Some implications of the sawtooth display are as follows:

[0080] • If a new pose knot is inserted between two existing pose knots, all the knots following the new pose knot will flip vertically so that the sawtooth is preserved.

[0081] • Tangent handles, if displayed, must be flipped only on those segments of the spline that are flipped. This can make the tangents appear broken. This is addressed by not allowing the user to break the tangents when in sawtooth mode.

[0082] Both pose knots (representing the poses of the animated object) and timing knots (representing the timing between poses) can be inserted at selected points in a spline, subject to certain constraints based on the information entry mode of interest. Pose knots can be inserted through direct manipulation or in a table of variables describing the image, but the pose knots are not insertable in a display. Timing knots can be inserted in a display or in a table. Moreover, a timing knot can be transformed into a pose knot. In split, the value of a knot represents the distance between poses and the timing knots represent the timing between the poses. The timing knots can never extend higher or lower than associated pose knots, since the only represent timing changes between poses. Likewise, pose knots can be moved in time (left or right on Fig. 2 or 3) but not in value. These features and limitations are primarily of interest to the animator user in employing the invention in creative effort, but they are not necessary limitations of the invention.

[0083] Because the present system provides an automated and integrated functionality, the inventors believe that the transformations performed by this system are much more consistent from frame to frame. Further, the inventors believe that the transformation of objects performed by this system provide a more accurate representation of real-world physics. Accordingly, the inventors believe that frames of animation including objects processed by the present system will be noticeably more realistic that was previously obtainable.

[0084] The invention has been explained with respect to specific embodiments. Other embodiments will be evident to those of ordinary skill in the art. Thus, the invention is not to be considered limited except as indicated by the appended claims.